Automatic Single-Image-Based Rain Streaks Removal via Image Decomposition

Li-Wei Kang, Member, IEEE, Chia-Wen Lin, Senior Member, IEEE, and Yu-Hsiang Fu

Abstract—Rain removal from a video is a challenging problem and has been recently investigated extensively. Nevertheless, the problem of rain removal from a single image was rarely studied in the literature, where no temporal information among successive images can be exploited, making the problem very challenging. In this paper, we propose a single-image-based rain removal framework via properly formulating rain removal as an image decomposition problem based on morphological component analysis (MCA). Instead of directly applying conventional image decomposition technique, the proposed method first decomposes an image into the low-frequency and high-frequency parts using a bilateral filter. The high-frequency part is then decomposed into a “rain component” and a “non-rain component” by performing dictionary learning and sparse coding. As a result, the rain component can be successfully removed from the image while preserving most original image details. Experimental results demonstrate the efficacy of the proposed algorithm.

Index Terms—Rain removal, sparse representation, dictionary learning, image decomposition, morphological component analysis (MCA).

I. INTRODUCTION

DIFFERENT weather conditions such as rain, snow, haze, or fog will cause complex visual effects of spatial or temporal domains in images or videos [1]–[11]. Such effects may significantly degrade the performances of outdoor vision systems relying on image/video feature extraction [12]–[17] or visual attention modeling [18], such as image registration [10], event detection [9], object detection [15]–[17], tracking, and recognition, scene analysis [18] and classification, image indexing and retrieval [12], and image copy/near-duplicate detection. A comprehensive survey of detection approaches for outdoor environmental factors, such as rain and snow, to enhance the accuracy of video-based automatic incident detection systems can be found in [9].

Removal of rain streaks has recently received much attention [2]–[4], [6]. To the best of our knowledge, current approaches are all based on detecting and removing rain streaks in a video. This work is among the first to specifically address the problem of removing rain streaks in a single image. Note, rain removal in an image may also fall into the category of the problem about image noise removal or image restoration. Hence, in the following subsections, we first briefly review current vision-based (video-based) rain removal approaches and image noise removal, followed by presenting our motivations of single-image-based rain streak removal and contribution of the proposed method.

A. Vision-based Rain Detection and Removal

Removal of rain streaks has recently received much attention. A pioneering work on detecting and removing rain streaks in a video was proposed in [2], where the authors developed a correlation model capturing the dynamics of rain and a physics-based motion blur model characterizing the photometry of rain. It was subsequently shown in [3] that some camera parameters, such as exposure time and depth of field can be selected to mitigate the effects of rain without altering the appearance of the scene. Moreover, an improved video rain streak removal algorithm incorporating both temporal and chromatic properties was proposed in [6]. The method proposed in [7] further utilizes the shape characteristics of rain streak for identifying and removing rain streaks from videos. Furthermore, a model of the shape and appearance of a single rain or snow streak in the image space was developed in [1] to detect rain or snow streaks. Then, the amount of rain or snow in the video can be reduced or increased. In [8], selection rules based on photometry and size are proposed to select the potential rain streaks in a video, where a histogram of orientations of rain streaks, estimated with geometric moments, is computed.

Moreover, some research works [10], [11] focus on raindrop detection in images or videos (usually on car windshields) which is different from the detection of rain streaks. A video-based raindrop detection method for improving the accuracy of image registration was proposed in [10], where a photometric raindrop model was utilized to perform monocular raindrop detection in video frames. In addition, a detection method for detecting raindrops on car windshields using geometric-photometric environment construction and intensity-based correlation was proposed in [11], which can be applied to vision-based driver assistance systems.
B. Image Noise Removal

Image noise removal or denoising problem is important and challenging [19]. The major goal of image noise removal is to design an algorithm that can remove unstructured or structured noise from an image which is acquired in the presence of an additive noise. Numerous contributions for image denoising in the past 50 years addressed this problem from many and diverse points of view. For example, spatial adaptive filters, stochastic analysis, partial differential equations, transform-domain methods, splines, approximation theory methods, and order statistics are some of the directions explored to address this problem [20].

Recently, the use of sparse and redundant representations over learned dictionaries has become one specific approach towards image denoising, which has proven to be effective and promising [20]. Based on the assumption that image signals admit a sparse decomposition over a redundant dictionary, by using the K-SVD dictionary training algorithm [21], Elad and Aharon [20] obtained a dictionary describing the image content effectively. They proposed two training options, where one is using the corrupted image itself and the other one is training on a set of high-quality images. They have shown how such Bayesian treatment leads to a simple and effective denoising algorithm, which achieves state-of-the-art image denoising performance. Moreover, similar idea has been successfully extended to solve more general image restoration problems, such as removing nonhomogeneous noise or recovering missing information (e.g., text removal and inpainting [22], [23] and binary artifacts removal from video-game images [24]). Although these dictionary-based image denoising scheme can also be used for removing rain streaks, they usually cannot do a good job in rain removal as will be shown in Sec. V.

C. Motivations of Single-Image-Based Rain Streak Removal

So far, the research works on rain streak removal found in the literature have been mainly focused on video-based approaches that exploit temporal correlation in multiple successive frames. Nevertheless, when only a single image is available, such as an image captured from a digital camera/camera-phone or downloaded from the Internet, a single-image based rain streak removal approach is required, which was rarely investigated before. In addition, some video rain removal approaches [3] based on adjusting camera parameters may not be suitable to consumer camcorders [6] and cannot be applied to existing acquired image/video data. Furthermore, for removing rain streaks from videos acquired from a moving camera, the performances of existing video-based approaches may be significantly degraded. The reason is that, since these video-based approaches usually perform rain streak detection, followed by interpolating the detected pixels affected by rain streaks in each frame, the non-stationary background due to camera motions and inaccurate motion estimation caused by the interference of rain streaks would degrade the accuracy of video-based rain streak detection and pixel interpolation. Even though some camera motion estimation techniques can be applied first to compensate for the camera motions [6], its performance may also be degraded by rain streaks or large moving activity. Moreover, for the case of steady effects of rain, i.e., pixels may be affected by rain across multiple consecutive frames, it is hard to detect these pixels or find reliable information from neighboring frames to recover them [2].

Moreover, many image-based applications such as mobile
visual search [12], object detection/recognition, image registration, image stitching, and salient region detection heavily rely on extraction of gradient-based features that are rotation- and scale-invariant. Some widely-used features (descriptors) such as SIFT (scale-invariant feature transform) [13], SURF (speeded up robust features) [14], and HOG (histogram of oriented gradients) [15–17] are mainly based on computation of image gradients. The performances of these gradient-based feature extraction schemes, however, can be significantly degraded by rain streaks appearing in an image since the rain streaks introduce additional time-varying gradients in similar directions. For example, as illustrated in Fig. 1, the additional unreliable interesting points caused by rain streaks degrade the invariant properties of SIFT/SURF and lead to potentially erroneous image matching in related applications.

As an example shown in Fig. 2, we applied the HOG-based pedestrian detector released from [17] to the rain image shown in Fig. 2(a) and its rain-removed version (obtained by the proposed method presented in Sec. III) shown in Fig. 2(b), respectively. It can be found that the detection accuracy for the rain-removed version is better. In addition, visual attention models [18] compute a saliency map topographically encoding for saliency at each location in the visual input that simulates which elements of a visual scene are likely to attract the attention of human observers. Nevertheless, the performances of the model for related applications may also be degraded if rain streaks directly interact with the interested target in an image. Therefore, single-frame-based rain streak removal is desirable.

D. Contribution of Proposed Method

It should be noted that separating and removing rain streaks from the non-rain part in a single frame is not a trivial work as rain streaks are usually highly mixed with the non-rain part, making the decomposition of non-rain part very challenging. In this paper, we propose a single-image-based rain streak removal method by formulating rain streak removal as an image decomposition problem based on MCA [26–30]. In our method, an image is first decomposed into the low-frequency and high-frequency parts using a bilateral filter. The high-frequency part is then decomposed into “rain component” and “non-rain component” by performing dictionary learning and sparse coding based on MCA. The major contribution of this paper is three-fold: (i) to the best of our knowledge, our method is among the first to achieve rain streak removal while preserving geometrical details in a single frame, where no temporal or motion information among successive images is required; (ii) we propose the first automatic MCA-based image decomposition framework for rain streak removal; and (iii) the learning of the dictionary for decomposing rain steaks from an image is fully automatic and self-contained, where no extra training samples are required in the dictionary learning stage. Besides, the proposed method also offers another option in dictionary learning by collecting exemplar patches from a set of non-rain training images to learn an extended dictionary to enrich the dictionary, as detailed in Sec. III-C.

The rest of this paper is organized as follows. In Sec. II, we briefly review the concepts of MCA-based image decomposition, sparse coding, and dictionary learning techniques. Sec. III presents the proposed single-image-based rain streak removal framework. In Sec. IV, experimental results are demonstrated. Finally, Sec. V concludes this paper.

II. MCA-BASED IMAGE DECOMPOSITION, SPARSE CODING, AND DICTIONARY LEARNING

The key idea of MCA is to utilize the morphological diversity of different features contained in the data to be decomposed and to associate each morphological component to a dictionary of atoms. In this section, the conventional MCA-based image decomposition approaches [26–30], sparse coding [32], and dictionary learning [21, 33] techniques are briefly introduced. The symbols used in this paper are listed in Table I.

A. MCA-based Image Decomposition

Suppose that an image \(I\) of \(N\) pixels is a superposition of \(S\) layers (called morphological components), denoted by

\[
I = \sum_{s=1}^{S} I_s,
\]

where \(I_s\) denotes the \(s\)-th component, such as the geometric or textural component of \(I\). To decompose the image \(I\) into \(\{I_s\}_{s=1}^{S}\), the MCA algorithms [26–30] iteratively minimize the following energy function:

\[
E(\{I_s\}_{s=1}^{S}, \{\theta_s\}_{s=1}^{S}) = \frac{1}{2} \sum_{s=1}^{S} \left\| I_m - I_s \right\|_2^2 + \tau \sum_{s=1}^{S} E_g(I_s, \theta_s),
\]

where

\[
E_g(I_s, \theta_s) = \frac{1}{2} \sum_{u=1}^{N} \sum_{v=1}^{N} \left( g(u,v) - I_s(u,v) \right)^2.
\]
where $\theta \in \mathbb{R}^{m_s}$ denotes the sparse coefficients corresponding to $I_s$ with respect to dictionary $D_s$, $\tau$ is a regularization parameter, and $E_s$ is the energy defined according to the type of $D_s$ (denoted by $D_{gs}$ for a global dictionary or by $D_{ls}$ for a local dictionary). For a global dictionary $D_{gs} \in \mathbb{R}^{N \times m_s}, N \leq m_s$, the energy function $E_s$ is defined as

$$ E_s(I_s, \theta_s) = \frac{1}{2} \|I_s - D_{gs}\theta_s\|_2^2 + \lambda\|\theta_s\|_1, \quad (2) $$

where $\lambda$ is a regularization parameter. Usually, to decompose an image into its geometric and textural components, traditional basis functions, such as wavelets or curvelets, are used as the dictionary for representing the geometric component whereas global DCT (discrete cosine transform) basis functions are used as the dictionary for representing the textural component of the image [26]–[30].

With respect to a local dictionary $D_{ls} \in \mathbb{R}^{n \times m_s}, n \leq m_s$, $\theta^k_s \in \mathbb{R}^{n}$ represents the sparse coefficients of patch $b^k_s \in \mathbb{R}^n, k = 1, 2, ..., N$, extracted from $I_s$. Each patch $b^k_s$ can be extracted centralized with a pixel of $I_s$ and overlapped with adjacent patches. The energy function $E_s$ for the local dictionary can be defined as

$$ E_s(I_s, \theta_s) = \sum_{k=1}^{N} \left(\frac{1}{2} \|b^k_s - D_{ls}\theta^k_s\|_2^2 + \lambda\|\theta^k_s\|_1\right), \quad (3) $$

Usually, a local dictionary for representing the textural component of an image is either composed of traditional basis functions, such as local DCT [26]–[28], [30], or constructed from the dictionary learning procedure [29] described in Sec. II-B.

The MCA algorithms solve (1) by iteratively performing for each component $I_s$ the following two steps: (i) update of the sparse coefficients: this step performs sparse coding to solve $\theta_s$ or $\{\theta^k_s\}_{k=1}^{N}$ to minimize $E_s(I_s, \theta_s)$ while fixing $I_s$; and (ii) update of the components: this step updates $I_s$ or $\{b^k_s\}_{k=1}^{N}$ while fixing $\theta_s$ or $\{\theta^k_s\}_{k=1}^{N}$.

More specifically, in the case of decomposing $I$ into two components $I_s$, $s = 1, 2$, a key step of MCA is to properly select a dictionary built by combining two sub-dictionaries $D_{ls}, s = 1, 2$, $D_1$ and $D_2$ can be either global or local dictionaries and should be mutually incoherent, that is, $D_1$ can provide sparse representation for $I_1$ but not for $I_2$, and vice versa. To decompose $I$ into geometric ($I_1$) and textural ($I_2$) components, global wavelet or global curvelet is used as $D_1$, whereas global DCT or local DCT is used as $D_2$ in [26]–[28], [30]. A comprehensive description of dictionary selections and related parameter settings for different kinds of image decomposition can be found in Table 2 of [27]. On the other hand, in [29], a global wavelet/curvelet basis is also used as $D_1$, whereas $D_2$ is constructed through a local dictionary learning process described below. Finally, to decompose an image into two components, both $D_1$ and $D_2$ are required to sparsely represent each component individually, as illustrated in Fig. 3 for the proposed single-image-based rain streak removal. It should be noted that we do not directly apply (1)–(3) to solve the rain streak removal problem. The major differences between the proposed framework and traditional MCA-based approaches are described in Sec. III-A. More details about traditional MCA methods, such as parameter settings, can be found in [26]–[30].

B. Sparse Coding and Dictionary Learning

Sparse coding [31], [32] is a technique of finding a sparse representation for a signal with a small number of nonzero or significant coefficients corresponding to the atoms in a dictionary [21], [33]. The pioneering work in sparse coding proposed by Olshausen and Field [31] states that the receptive fields of simple cells in mammalian primary visual cortex can be characterized as being spatially localized, oriented, and bandpass. It was shown in [31] that a coding strategy that maximizes sparsity is sufficient to account for these three properties, and that a learning algorithm attempting to find sparse linear codes for natural scenes will develop a complete family of localized, oriented, and bandpass receptive fields.

As mentioned previously, it is required to construct a dictionary $D_{ls}$ containing the local structures of textures for sparsely representing each patch $b^k_s$ extracted from the textural component $I_s$ of image $I$. In some applications, we may use a set of available training exemplars (similar to the patches extracted from the component we want to decompose)
\( y^k \in R^n, k = 1, 2, ..., P \) to learn a dictionary \( D_{ls} \) sparsifying \( y^k \) by solving the following optimization problem:

\[
\min_{D \in R^{m_s \times n}, \theta \in R^{m_s}} \sum_{k=1}^{P} \left( \frac{1}{2} \| y^k - D_{ls} \theta^k \|_2^2 + \lambda \| \theta^k \|_1 \right),
\]

where \( \theta^k \) denotes the sparse coefficients of \( y^k \) with respect to \( D_{ls} \) and \( \lambda \) is a regularization parameter. Equation (4) can be efficiently solved by performing a dictionary learning algorithm, such as K-SVD [21] or online dictionary learning [33] algorithms, where the sparse coding step is usually achieved via OMP (orthogonal matching pursuit) [32]. Finally, the image decomposition is achieved by iteratively performing the MCA algorithm to solve \( I_s \) (while fixing \( D_{ls} \)) described in Sec. II-A and the dictionary learning algorithm to learn \( D_{ls} \) (while fixing \( I_s \)) until convergence. The convergence of the MCA image decomposition algorithms has been proven in [29].

The proposed rain removal framework described in Sec. III uses two local dictionaries learned from the training patches extracted from the rain image itself to respectively decompose a rain image into its rain component and geometric (non-rain) component without using any global dictionary. The main reasons include (i) We do not assume or empirically decide any type of global dictionary for representing either of the rain and geometrical components in the rain image; (ii) Because the geometric component is usually highly mixed with rain streaks in some regions of the rain image, segmenting the image into local patches would be easier to extract rain patches that mainly contain rain streaks to facilitate self-learning of rain atoms; (iii) Since rain streaks in different local regions of an image often exhibit different characteristics, local-patch-based dictionary learning would usually learn rain atoms that better represent rain streaks than a global dictionary does.

III. PROPOSED RAIN STREAK REMOVAL FRAMEWORK

Fig. 3 shows the proposed single-image-based rain streak removal framework, in which rain streak removal is formulated as an image decomposition problem. In our method, the input rain image is first roughly decompose into the low-frequency (LF) part and the high-frequency (HF) part using the bilateral filter [34], [35], where the most basic information will be retained in the LF part while the rain streaks and the other edge/texture information will be included in the HF part of the image as illustrated in Figs. 4(a) and 4(b). Then, we perform the proposed MCA-based image decomposition to the HF part that can be further decomposed into the rain component [see Fig. 4(c)] and the geometric (non-rain) component [see Fig. 4(d)]. In the image decomposition step, a dictionary learned from the training exemplars extracted from the HF part of the image itself can be divided into two sub-dictionaries by performing HOG [15] feature-based dictionary atom clustering. Then, we perform sparse coding [32] based on the two sub-dictionaries to achieve MCA-based image decomposition, where the geometric component in the HF part can be obtained, followed by integrating with the LF part of the image to obtain the rain-removed version of this image as illustrated in Figs. 4(e) and 4(f). The detailed method shall be elaborated below.

A. Major Differences between Proposed Method and Traditional MCA-based Approaches

As mentioned in Sec. II, traditional MCA algorithms usually use a fixed global dictionary based on wavelets/curvelets to represent the geometric component of an image. To represent the textural component of an image, either a fixed global (global DCT) or a local (local DCT) dictionary is used. In addition, a learned dictionary may also be used to represent the textural component. Nevertheless, to decompose an image into the geometric and textural components, the selection of dictionaries and related parameter tuning seems to be heavily empirical, as the examples shown in Table 2 of [27]. Based on our experience, it is not easy to select a proper fixed dictionary to represent rain streaks due to its variety.

Moreover, learning a dictionary for representing textural component usually assumes that a set of exemplar patches for
the texture to be represented can be either known in advance or extracted from an image to be decomposed itself. Nevertheless, in practice, it is usually not easy to select correct rain patches in a single rain image automatically. It is also not easy to directly extract pure rain patches for dictionary learning from a rain image due to that rain streaks usually cover most regions in a rain image. That is, the geometric and rain components are usually largely mixed. Moreover, even though a traditional fixed global dictionary based on wavelets/curvelets can well sparsely represent the geometric component of an image, using a learned dictionary based on the exemplar patches extracted from the component itself would be much better [38].

Therefore, rather than using a fixed dictionary, assuming prior training exemplar patches available, or resorting to tuning parameters for the used dictionary, our method extracts a set of selected patches from the HF (high-frequency) part of a rain image itself to learn a dictionary. Then, based on the features extracted from individual atoms, we classify the atoms constituting the dictionary into two clusters to form two sub-dictionaries for representing the geometric and rain components of the image, respectively. The dictionary learning process in the proposed method is elaborated in Sec III-C.

Traditional MCA algorithms are all directly performed on an image in the pixel domain. However, it is typically not easy to directly decompose an image into its geometric and rain components in the pixel domain, because the geometric and rain components are usually largely mixed in a rain image. This makes the dictionary learning process difficult to clearly identify the “geometric (non-rain) atoms” and “rain atoms” from the pixel-domain training patches with mixed components. This may lead to removing too many image contents that belong to the geometric component but are erroneously classified to the rain component.

Therefore, we propose to first roughly decompose a rain image into the low-frequency (LF) part and the HF part. Obviously, the most basic information of the image is retained in the LF part whereas the rain component and the other edge/texture information are mainly included in the HF part. The decomposition problem can be therefore converted to decomposing the HF part into the rain and other textural components. Such decomposition aids in the dictionary learning process as it is easier to classify in the HF part “rain atoms” and “non-rain atoms” into two clusters based on some specific characteristics of rain streaks.

Furthermore, traditional MCA-based image decomposition approaches are all achieved by iteratively performing the MCA algorithm and the dictionary learning algorithm until convergence. In contrast, the proposed method is non-iterative except for that the utilized dictionary learning, clustering, and sparse coding tools are essentially iterative, as will be explained below.

B. Preprocessing and Problem Formulation

For an input rain image $I$ in the preprocessing step, we apply a bilateral filter [34] to roughly decompose $I$ into the LF part ($I_{LF}$) and HF part ($I_{HF}$), i.e., $I = I_{LF} + I_{HF}$. The bilateral filter can smooth an image while preserving edges, by means of a nonlinear combination of nearby image values. In this step, we adjust the strength of smoothness of the bilateral filter to remove all of the rain streaks from $I$, as an illustrative example shown in Figs. 4(a) and 4(b). Then, our method learns a dictionary $D_{HF}$ based on the training exemplar patches extracted from $I_{HF}$ to further decompose $I_{HF}$, where $D_{HF}$ can be further divided into two sub-dictionaries, $D_{HF,G}$ and $D_{HF,R}$ ($D_{HF} = [D_{HF,G} | D_{HF,R}]$), for representing the geometric and rain components of $I_{HF}$, respectively. As a result, we formulate the problem of rain streak removal for image $I$ as a sparse coding-based image decomposition problem as follows:

$$\min_{\theta^k_{HF} \in \mathbb{R}^m} \|b^k_{HF} - D_{HF,\theta^k_{HF}}\|_2^2 \quad \text{s.t.} \quad \|\theta^k_{HF}\|_0 \leq L,$$

where $b^k_{HF} \in \mathbb{R}^n$ represents the $k$-th patch extracted from $I_{HF}$, $k = 1, 2, ..., P$. $\theta^k_{HF} \in \mathbb{R}^m$ are the sparse coefficients of $b^k_{HF}$ with respect to $D_{HF} \in \mathbb{R}^{n \times m}$, $n \leq m$, and $L$ denotes the sparsity or maximum number of nonzero coefficients of $\theta^k_{HF}$. Since $l_0$-minimization is hard to optimize, one usually solves the $l_1$-minimization problem which in most cases give comparable results [36], [37]. Therefore, solving the $l_0$-minimization problem in (5) can be cast to solve the following $l_1$-minimization problem:

$$(\theta^k_{HF})^* = \arg \min_{\theta^k_{HF} \in \mathbb{R}^m} \left( \frac{1}{2} \|b^k_{HF} - D_{HF,\theta^k_{HF}}\|_2^2 + \lambda \|\theta^k_{HF}\|_1 \right),$$

where $(\theta^k_{HF})^*$ denotes the solution minimizing (6) and $\lambda$ is a regularization parameter. To solve (5), we apply the efficient OMP implementation provided in [33]. Each patch $b^k_{HF}$ can be reconstructed and used to recover either the geometric or rain component of $I_{HF}$ depending on the corresponding nonzero coefficients in $\theta^k_{HF}$, i.e., the used atoms from $D_{HF,G}$ or $D_{HF,R}$.

C. Dictionary Learning and Partition

1) Dictionary Learning: In this step, we extract from $I_{HF}$ a set of overlapping patches as the training exemplars $y^k$ for learning dictionary $D_{HF}$. We formulate the dictionary learning problem as [21], [33]

$$\min_{D_{HF} \in \mathbb{R}^{n \times m}, \theta^k \in \mathbb{R}^m} \frac{1}{P} \sum_{k=1}^{P} \left( \frac{1}{2} \|y^k - D_{HF,\theta^k}\|_2^2 + \lambda \|\theta^k\|_1 \right).$$

where $\theta^k$ denotes the sparse coefficients of $y^k$ with respect to $D_{HF}$ and $\lambda$ is a regularization parameter. In this work, we apply an efficient online dictionary learning algorithm proposed in [33] to solve (6) to obtain $D_{HF}$, as illustrated in Fig. 5.

2) Dictionary Partition and Identification: We find that the atoms constituting $D_{HF}$ can be roughly divided into two clusters (sub-dictionaries) for representing the geometric and rain components of $I_{HF}$, respectively. Intuitively, the most significant feature for a rain atom can be extracted via “image gradient.” In the proposed method, we utilize the HOG descriptor [15] to describe each atom in $D_{HF}$. The HOG method [15] is briefly introduced as follows.

The basic idea of HOG is that local object appearance and shape can be usually well characterized by the distribution of local intensity gradients or edge directions, without precisely
knowing the corresponding gradient or edge positions [15]. To extract the HOG feature from an image, the image can be divided into several small spatial regions or cells. For each cell, a local 1-D histogram of gradient directions or edge orientations over the pixels of the cell can be accumulated. The combined histogram entries of all cells form the HOG representation of the image. In our implementation, the size of a local image patch/dictionary atom is chosen to be 16×16, which leads to reasonable computational cost in dictionary partition (involving HOG feature extraction) as will be shown in Sec. IV.

After extracting the HOG feature for each atom in $D_{HF}$, we then apply the K-means algorithm to classify all of the atoms in $D_{HF}$ into two clusters $D_1$ and $D_2$ based on their HOG feature descriptors. The following procedure is to identify which cluster consists of rain atoms and which cluster consists of geometric or non-rain atoms. First, we calculate the variance of gradient direction for each atom $d_{ij}, j = 1, 2, ..., N$, in cluster $D_i$, as $VG_{ij}$, where $N_i$ denotes the number of atoms in $D_i, i = 1, 2$. Then, we calculate the mean of $VG_{ij}$ for each cluster $D_i$ as $MVG_i$. Based on the fact that the edge directions of rain streaks in an atom are usually consistent, i.e., the variance of gradient direction for a rain atom should be small, we identify the cluster with the smaller $MVG_i$ as rain sub-dictionary $D_{HF,R}$, and the other one as geometric (or non-rain) sub-dictionary $D_{HF,G}$, as depicted in Fig. 6.

On the other hand, although the dictionary learning step in the proposed method can be fully self-contained, where no extra training samples are required, the decomposition performance can be further improved by collecting a set of exemplar patches from the HF parts of some training non-rain images to learn an extended global dictionary $D_E$ to enrich the dictionary. Fig. 7 illustrates an example of $D_E$. Then, we integrate $D_{EG}$ with $D_{HF,G}$ of each image to form the final geometric sub-dictionary of the image.

Moreover, based on our experiences, it is hard to learn a rain dictionary by collecting a set of real exemplar rain patches due to the following reasons: (i) it is not easy to collect pure rain patches extracted from natural rain images because rain streaks are usually highly mixed with the non-rain part in an image; (ii) it is also not easy to learn a dictionary adapted to a wide range of lighting and viewing conditions for rain streaks; and (iii) even though a few photorealistic rendering techniques have been proposed [5], it is still not easy to synthesize all possible real rain patches for learning a representative rain dictionary adapted to a large variety of rain streaks. Hence, we proposed a self-learning approach that learns the rain dictionary for a rain image based on a set of exemplar patches extracted from the image itself, followed by performing dictionary partition. The rain dictionary learned from an image itself is more appropriate to sparsely represent the rain component of the image.

3) Diversities of Two Sub-dictionaries: The MCA algorithms distinguish between the morphological components by taking advantage of the diversities of two dictionaries $D_1$ and $D_2$, which can be measured by the mutual incoherence of them [28]. The mutual coherence $\mu(D_1, D_2)$ between $D_1$ and $D_2$ can be
where $d_{ij}$ and $d_{lj}$ stand for the $i$-th and $j$-th atoms (rearranged as a column vector) in $D_{IF}$ and $D_{F}$, respectively, and $(d_{ij}, d_{lj})$ denotes the inner product of $d_{ij}$ and $d_{lj}$. When each atom is normalized to have a unit $l_2$-norm, the range of $\mu(D_{ij}, D_{lj})$ is $[0, 1]$. As a result, the mutual incoherence is $1 - \mu(D_{ij}, D_{lj})$. The smaller the mutual coherence is, the larger the diversities of the two sub-dictionaries will be, and thus the better the decomposition performance based on the two dictionaries will be. The experimental evaluations of the mutual incoherence of rain sub-dictionary $D_{IF,R}$ and geometric sub-dictionary $D_{IF,G}$ for decomposing a rain image in the proposed method are presented in Sec. IV.

D. Removal of Rain Streaks

Based on the two dictionaries $D_{IF,R}$ and $D_{IF,G}$, we perform sparse coding by applying the OMP (orthogonal matching pursuit) algorithm [32] for each patch $b_{IF}^k$ extracted from $I_{IF}$ via minimization of (5) to find its sparse coefficients $\hat{b}_{IF}^k$. Different from traditional MCA algorithms, where the sparse coding and dictionary learning should be iteratively performed, we perform sparse coding only once for each patch $b_{IF}^k$ with respect to $D_{IF} = [D_{IF,G} D_{IF,R}]$.

Then, each reconstructed patch $b_{IF}^k$ can be used to recover either geometric component $I_{G,R}^k$ or rain component $I_{F,R}^k$ of $I_{IF}$ based on the sparse coefficients $\hat{b}_{IF}^k$ as follows. We set the coefficients corresponding to $D_{IF,G}$ in $\hat{b}_{IF}^k$ to zeros to obtain $\hat{b}_{IF,R}^k$, while the coefficients corresponding to $D_{IF,R}$ in $\hat{b}_{IF}^k$ to zeros to obtain $\hat{b}_{IF,G}^k$. Therefore, each patch $b_{IF}^k$ can be re-expressed as either $\hat{b}_{IF,R}^k = D_{IF,G} \times \hat{b}_{IF,G}^k$ or $\hat{b}_{IF,G}^k = D_{IF,R} \times \hat{b}_{IF,R}^k$, which can be used to recover $I_{G,R}^k$ or $I_{F,R}^k$, respectively, by averaging the pixel values in overlapping regions. Finally, the rain-removed version of the image $I$ can be obtained via $I_{Non,Rain} = I_{LF} + I_{IF}^R$, as illustrated in Fig. 4(e). In summary, the proposed single-image-based rain streak removal method is summarized in Table II.

IV. EXPERIMENTS AND DISCUSSION

A. Performance Evaluation

Because we cannot find any other single-frame-based approach, to evaluate the performance of the proposed algorithm, we first compare the proposed method with a low-pass filtering method called the bilateral filter proposed in [34], which has been extensively applied and investigated recently for image processing, such as image denoising [35]. Besides, to demonstrate that existing image denoising methods cannot well address the problem of single-image-based rain removal, we also compare the proposed method with the state-of-the-art image denoising method based on K-SVD dictionary learning and sparse representation proposed in [20] with a released source code available from [23] (denoted by “K-SVD-based denoising”).

To the best of our knowledge, no standard still rain image dataset is currently available for benchmarking. Existing video-based rain removal approaches [1]–[3], [6], [7] were all evaluated by collecting a few video frames filmed by the authors or extracted from existing movie files. Hence, we collected several natural/synthetic rain images from the Internet and also from the photo-realistically rendered rain video frames (with ground-truth images) provided in [5] for a few of them. For natural rain images, it is not easy to provide many quantitative analyses due to the fact that the ground-truth images are usually unavailable. In current video-based rain removal research [2], [3], [6], [7], the performances were usually subjectively evaluated. On the other hand, to evaluate the quality of a rain-removed image with a ground-truth, we used the visual information fidelity (VIF) metric [39] in the range of $[0, 1]$ which has been shown to outperform PSNR (peak signal-to-noise ratio) metric. More test results can be found in our project website [42], where our test image dataset can be downloaded.

The synthesized rain images shown in Figs. 1(b) [40] and 8(b) [41] were generated by adding rain streaks to Figs. 1(a) and 8(a), respectively, using the Photoshop software [40], [41]. On the other hand, the rendered rain images shown in Figs. 9–11 were generated by photorealistic rendering technique proposed in [5], briefly described as follows. In [5], a rain streak appearance model capturing the complex interactions between the lighting direction, viewing direction, and oscillating shape of a raindrop was proposed. This model is built upon a raindrop oscillation model which was developed in atmospheric sciences. The oscillation parameters were empirically decided by measuring rain streak appearances under a wide range of lighting and

| TABLE II PROPOSED SINGLE-IMAGE-BASED RAIN STREAK REMOVAL ALGORITHM |
|-----------------|-----------------|
| Input: a rain image $I$. Output: the rain-removed version $I_{Non,Rain}$ of $I$. |
| 1. Apply the bilateral filter to obtain the LF part $I_{LF}$ and HF part $I_{IF}$, such that $I = I_{LF} + I_{IF}$. |
| 2. Extract a set of image patches $y^k \in \mathbb{R}^n$, $k = 1, 2, ..., P$, from $I_{IF}$. Apply the online dictionary learning for sparse coding algorithm to solve |
| \[
| \min_{d_{IF} \in \mathbb{R}^n; \theta \in \mathbb{R}^n} \frac{1}{P} \sum_{k=1}^{P} \frac{1}{2} \|y^k - D_{IF}\theta^k\|_2^2 + \lambda\|\theta\|_1 |
| \]
| to obtain the dictionary $D_{IF}$ consisting of the atoms that can sparsely represent $y^k$, $k = 1, 2, ..., P$. |
| 3. Extract HOG feature descriptor for each atom in $D_{IF}$. Apply K-means algorithm to classify all of the atoms into two clusters based on their HOG feature descriptors. |
| 4. Identify one of the two clusters as “rain sub-dictionary,” $D_{IF,R}$ and the other one as “geometric sub-dictionary,” $D_{IF,G}$.
| 5. Apply MCA by performing OMP to solve |
| \[
| \min_{b_{IF} \in \mathbb{R}^n} \|D_{IF} - D_{IF}\theta^k\|_2 \text{ s.t. } \|\theta^k\|_0 \leq L |
| \]
| for each patch $b_{IF}^k \in \mathbb{R}^n$, $k = 1, 2, ..., P$, in $I_{IF}$ with respect to $D_{IF} = [D_{IF,G} D_{IF,R}]$. |
| 6. Reconstruct each patch $b_{IF}^k$ to recover either geometric component $I_{G,R}^k$ or rain component $I_{F,R}^k$ of $I_{IF}$ based on the corresponding sparse coefficients obtained from Step 5. |
| 7. Return the rain-removed version of $I$ via $I_{Non,Rain} = I_{LF} + I_{IF}^R$. |

The performance evaluation results for both rain streaks and rain drops are shown in Figs. 9–11. The VIF and PSNR values of the rain-removed images $I_{Non,Rain}$ are summarized in Table II. In summary, the proposed single-image-based rain streak removal method is a potential candidate for real-time applications, especially when the computational resources and time are limited.
viewing conditions. Based on these parameters, rain streaks with variations in streak appearance with respect to lighting and viewing directions were rendered. An efficient image-based rendering algorithm was also proposed in [5] to add rain to an image or video, which requires a coarse depth map of the scene, and the locations and properties of the light sources. It should be noted that the proposed method does not assume that any knowledge about the rain streak appearance in a rain image can be available in advance.

Besides, we also compare our method with a video-based rain removal method based on adjusting camera parameters proposed in [3] (denoted by “video-based camera see”), which should outperform most of other video-based techniques without adjusting cameras. We captured some single frames from the videos released from [3] and compared our results with the ones of [3] from the same videos. For each video released from [3], the preceding frames are rain frames, followed by succeeding rain-removed frames in the same scene. We pick a single rain frame from the preceding frames for rain removal and compared our results with the rain-removed one [3] of a similar frame from the succeeding frames in the same video (no exactly the same frame is available for comparison).

The parameter settings of the proposed method are described as follows. The implementation of the bilateral filter is provided by [43], where we set the spatial-domain and intensity-domain standard deviations to 6 and 0.2, respectively, to ensure that most rain streaks in a rain image can be removed. In the dictionary learning step, we used an efficient implementation provided in [33] with the suggested regularization parameter λ used in (7) set to 0.15, which is also suggested by the sparse coding process performed in [24]. It should be noted that parameters λ’s used in (2)–(4), (6), and (7), have the same meaning, and hence, we used the same symbol (λ) for convenience. In fact, only (7) is used in the proposed method. For each test gray-scale image of size N × N (= 256 × 256 in our experiments), the patch size, number of training patches, dictionary size, and the number of training iterations are set to N = 16 × 16, P = (N − √N + 1) × (N − √N + 1), m = 1024, and 100, respectively. We also used the efficient OMP implementation provided in [33] with the number of nonzero coefficients set to at most 10 as suggested in [33], That is, L in (5) is set to 10. The smaller the value of L is, the sparser the solution of (5) becomes, and vice versa. A smaller value of L leads to lower computational complexity but fewer atoms in the dictionary, and vice versa. We evaluated several possible values of L and found that L = 10 achieves the best tradeoff in most test cases. The used HOG implementation is provided by [16] with the dimension of each feature descriptor set to 81. The number of iterations for K-means clustering is 100.

We also evaluate the performance of the proposed method with extended dictionary D_E that is integrated with the respective geometric sub-dictionary for each test image. We collected several training patches extracted from the HF parts of eight widely-used non-rain images, including Baboon, Barbara, F-16, Goldhill, House, Lena, Man, and Pepper images. The patch size, dictionary size, and number of training patches, dictionary size, and the number of training iterations are set to N = 256 × 256 in our experiments), the patch size, number of training iterations are set to N = 16 × 16, P = (N − √N + 1) × (N − √N + 1), m = 1024, and 100, respectively. We also used the efficient OMP implementation provided in [33] with the number of nonzero coefficients set to at most 10 as suggested in [33], That is, L in (5) is set to 10. The smaller the value of L is, the sparser the solution of (5) becomes, and vice versa. A smaller value of L leads to lower computational complexity but fewer atoms in the dictionary, and vice versa. We evaluated several possible values of L and found that L = 10 achieves the best tradeoff in most test cases. The used HOG implementation is provided by [16] with the dimension of each feature descriptor set to 81. The number of iterations for K-means clustering is 100.

We also evaluate the performance of the proposed method with extended dictionary D_E that is integrated with the respective geometric sub-dictionary for each test image. We collected several training patches extracted from the HF parts of eight widely-used non-rain images, including Baboon, Barbara, F-16, Goldhill, House, Lena, Man, and Pepper images. The patch size, dictionary size, and number of training

![Rain removal results: (a) the original non-rain image (ground-truth); (b) the rain image of (a); (c) the rain-removed version of (b) via the bilateral filter (VIF = 0.31); (d) the HF part of (b); (e) the rain sub-dictionary for (d); (f) the geometric sub-dictionary for (d); (g) the rain component of (d); (h) the geometric component of (d); (i) the rain-removed version of (b) via the proposed method (VIF = 0.53, μ = 0.7618); and (j) the rain-removed version of (b) via the proposed method with D_E (VIF = 0.57).](image)
iterations are set to 16×16, 1024, and 200, respectively. The $D_E$ learning process is offline performed only once. The eight training images and $D_E$ are shown in Fig. 7.

Besides, we compare our method with the K-SVD-based image denoising method [20], in which only one dictionary is used for sparse coding stage, based on the assumption that the standard deviation of noise, which is assumed to be Gaussian distributed, can be known in advance. We set the parameters used in the K-SVD method according to the suggestions in [20, 23], where the patch size, dictionary size, and the number of training iterations are set to 8×8, 256, and 15, respectively. In rain removal applications, the standard deviation value of the rain noise is usually unknown. To estimate the standard deviation of the rain noise, for a rendered rain image with a ground-truth, we directly calculate the deviation of the each rain component patch as an initial value, whereas for a natural rain image, we use the rain component obtained from the proposed method to estimate the initial value. We then manually tune the value around the initial value to ensure that most of the rain streaks in the rain image can be removed.

The rain removal results obtained from the bilateral filter [34], the K-SVD method [20], the proposed method, and the proposed method with $D_E$ are shown in Figs. 9–13, where the test images in Figs. 9–11 are rendered rain images provided in [5]. The VIF results for the test images are summarized in Table III. The simulation results demonstrate that although the bilateral filter and the K-SVD denoising filter can remove most rain streaks, they both simultaneously remove much image detail as well. The
proposed methods successfully remove most rain streaks while preserving most non-rain image details in these test cases, thereby improving the subjective visual quality significantly. It can be observed that, the K-SVD usually cannot do a good job in rain streaks removal due to the following two reasons. First, the dictionary learned from non-rain natural images usually contains atoms that can represent rain streaks sufficiently well as rain streaks have similar content with many of non-rain image details. As a result, the rain part would not be neglected by the dictionary when the sparsity constraint is imposed since there will be quite a few non-zero coefficients corresponding to these rain-like atoms, especially when the non-rain component is highly mixed with the rain streaks. Zeroing out smaller non-zero coefficients will remove both the rain part and the details of non-rain part, thereby resulting in a seriously blurred de-rained image as can be observed from the test results. The second reason is that, the K-SVD scheme assumes that some knowledge about the noise statistics is known in advance (e.g., the standard deviation of the noise for denoising [20], [23], or even the additional information about the location of the noise for overlay text removal or inpainting [23], [24]), so that the reconstruction error can be well bounded. However, the assumption is not valid for rain streaks removal applications, as it is difficult to obtain or estimate the information in such applications.

In contrast, considering that rain streaks are usually coherent in an image and still significantly differ from the geometrical component in most parts of the image, our method addresses the above-mentioned problems by using two separate dictionaries that learns the rain atoms and non-rain atoms from the input image itself for sparsely representing the rain and non-rain components of the image. Our approach further makes use of the fact that the input image contains the rain patches that are the best for training the rain atoms and the gradient features of rain patches in an image has similar statistics in terms of gradient magnitude and directions (e.g., the HOG features).

Moreover, the results obtained from the “video-based camera see” method [3] and proposed methods are compared in Fig. 14 (more results can be found in [42]). The simulation results demonstrate that, when rain streaks are obviously visible in a single frame, the proposed method achieves comparable visual quality with existing video-based methods without the need of using temporal information of successive frames and adjusting camera parameters.

It can be observed from Figs. 4 and 8–14 that, compared to the proposed methods without and with extended dictionary $D_E$, incorporating the extended geometric dictionary leads to better visual quality while increasing the computational complexity (see the run-time analysis in Table IV shown below) of sparse coding due to the much larger size of the extended dictionary. The reason why sparse coding with an extended geometric dictionary usually achieves better visual quality than that without extended dictionary is that the extended dictionary provides more non-rain atoms for sparse coding to recover rain-removed version with more image details. Note, incorporating $D_E$ in rain removal is only an option which leads to visual quality improvement with increased complexity in most test cases. In some rare cases, however, the extended dictionary $D_E$ may not improve the visual quality of some rain patches because using $D_E$ can possibly derive inharmonious textures in a rain-removed image. The main reason is that $D_E$ is
a much richer dictionary learned by several non-rain image patches and can be used to speculatively recover some texture information behind the rain streaks in the rain image while applying the MCA image decomposition. Note, the values of mutual coherence ($\mu$) between the two sub-dictionaries usually fall in the range of [0.6, 0.9], which is not very close to zero. The main reason is that the two sub-dictionaries used in the proposed method are generated from a single learned dictionary based on a single feature (HOG) based clustering. It is unavoidable that the two dictionaries may have few somewhat coherent atoms, which will dominate the $\mu$ value. In the literature reporting $\mu$ values, minimization of the $\mu$ between a sensing matrix and a fixed dictionary for learning an optimal sensing matrix was mentioned in [44]. In [44], some $t$-averaged $\mu$ values (approaching $\mu$ when $t$ grows) between two matrices were reported to be in a range of [0.4, 0.6], where one matrix is randomly initialized. Hence, based on the obtained rain removal results of our method and the comparison of the ranges of $\mu$ between our method and [44], the $\mu$ values of our method are usually small enough.

The proposed method was implemented in MATLAB® on a personal computer equipped with Intel® Core™ i5-460M processor and 4 GB memory. The run-time of each key step, including the bilateral filtering, dictionary learning, dictionary partition, and sparse coding (without and with $D_E$), for each test image (Figs. 8–11) is listed in Table IV. It can be found that the run-time of the dictionary learning step dominates the total run time, which may be further reduced for future work. In Table IV, we also indicate the memory usage of our method, which is mainly dominated by the memory used for the sparse coding dictionary. In our method without extended dictionary, the self-learned dictionary contains totally 1024 atoms where each atom consumes 16x16 bytes, leading to a memory usage of 256K bytes. Besides, the extended dictionary consumes additional 1024 atoms, thereby requiring 512K bytes in total if the extended dictionary is utilized.

### B. Discussion

Besides collecting training exemplar patches from some training non-rain images for learning $D_{E_c}$, we may extract training patches from the same or neighboring camera(s) when extending the proposed method to video rain removal. That is,
we may extract exemplar patches from the neighboring rain-removed frames captured by intra/inter cameras in the same scene. Then, we can integrate the geometric sub-dictionary obtained from the HF part itself and the extended global dictionary learned from the pre-collected training patches to form the final geometric sub-dictionary.

Moreover, the shared-private factorization scheme proposed in [45] may be used to further improve the performance of image decomposition. In [45], to best leverage the information contained in each view of an image represented by multiple views/modalities, inspired by structured spare coding, the authors proposed an approach to learning factorized representations of multi-view data in which the information is correctly factorized into components that are shared across several views and private to each view. The concept of shared-private factorization [45] may be applied to further improve our work in two aspects. First, a rain image can be segmented into several local regions with different local characteristics, which can be viewed as the multiple views. We may apply the multi-view learning to learn a latent space that can separate the information (rain atoms) shared among several views from the information (unique non-rain atoms) private to each view. Then, the two dictionaries for rain removal can be identified accordingly. Second, rather than learning two disjoint private dictionaries without any common atoms for the rain and non-rain components, the two dictionaries may share some common atoms (i.e., the shared dictionary). Soft clustering (e.g., the fuzzy C-means) rather than hard clustering, or shared-private factorization, can be used to obtain better sparse representations.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a single-image-based rain streak removal framework by formulating rain removal as an MCA-based image decomposition problem solved by performing sparse coding and dictionary learning algorithms. The dictionary learning of the proposed method is fully automatic and self-contained where no extra training samples are required in the dictionary learning stage. We have also provided an optional scheme to further enhance the performance of rain removal by introducing an extended dictionary of non-rain atoms learned from non-rain training images. Our experimental results show that the proposed method achieves comparable performance with state-of-the-art video-based rain removal algorithms without the need of using temporal or motion information for rain streak detection and filtering among successive frames.

For future work, the performance of our method may be further improved in terms of computational complexity and visual quality by enhancing the sparse coding, dictionary learning, and dictionary partitioning processes. More specifically, since the dictionary learning and sparse coding consume most of execution time, the input image can be segmented into several local regions with different local characteristics such that the on-line dictionary learning for individual regions can be performed in parallel to accelerate the two processes by taking advantage of current multi-core processor technology. Besides, with the localized learning and sparse coding, the number of patches of each local region for dictionary learning and the number of atoms for sparse coding will be significant fewer than those for the whole-image-based learning, thereby further reducing the computational complexities of the two processes. Nevertheless, the impact of localized learning and sparse coding on rain removal performance needs in-depth investigation. Moreover, the dictionary learning process can be further improved to obtain more accurate sparse representations by taking into account the information shared in rain and non-rain components, and the information shared in the rain components of different local regions as mentioned in Sec. IV.

REFERENCES


